

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/839,679
Filing Date April 20, 2001
Inventorship Sullivan et al.
Applicant Microsoft Corp.
Group Art Unit 2621
Examiner Czekaj
Attorney's Docket No. msl-601
Title: "An Application Program Interface (API) Facilitating Decoder Control of
Accelerator Resources"

APPEAL BRIEF

To: Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Rich Bucher (Tel. 509-324-9256x216; Fax 509-323-8979)
Customer No. 22801

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits an appeal brief for application 09/839,679, filed April 20, 2001, within the requisite time from the date of filing the Notice of Appeal. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<u>Appeal Brief Items</u>	<u>Page</u>
(1) Real Party in Interest	3
(2) Related Appeals and Interferences	3
(3) Status of Claims	3
(4) Status of Amendments	3
(5) Summary of Claimed Subject Matter	4
(6) Grounds of Rejection to be Reviewed on Appeal	5
(7) Argument	6
(8) Appendix of Appealed Claims	20
(9) Evidence appendix	26
(10) Related Proceedings appendix	27

(1) Real Party in Interest

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject invention.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1-25 stand rejected and are pending in the Application. Claims 1-25 are set forth in the Appendix of Appealed Claims on page 20.

(4) Status of Amendments

A first Office Action was mailed August 24, 2004.

A Response was filed October 7, 2004. No claims were amended.

A Final Office Action was mailed January 26, 2005. No claims were amended.

A non-final Office Action was mailed July 13, 2005.

A response was filed October 26, 2005. Claims 1, 12, and 18 were amended.

A Final Office Action was mailed March 24, 2006.

A Notice of Appeal was filed on May 24, 2006.

(5) Summary of Claimed Subject Matter

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters, if any. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed subject matter and the claims are not limited to solely the elements corresponding to these reference characters.

With regard to claim 1, a method comprising: receiving a command from a decoder application (Fig. 2 (160A-N)) at an application program interface (API) (Fig. 2 (104)), wherein the API is configured to facilitate the use of a plurality of different multimedia accelerators (Fig. 2 (174A-N)) with the decoder application (Page 15, lines 2-18; Fig. 2 (104, 160A-N, 174 A-N)); and generating one or more filter control command data structures (Fig. 2 (204); Page 52, line 19 through page 53, line 2; Fig. 3 (300)), recognizable by a communicatively coupled accelerator including one or more parameters which, when received by the accelerator, affects one or more filter settings of the accelerator based, at least in part, on the content of the received command (Page 23, lines 9-25; Fig. 2 (204)).

With regard to claim 12, a storage medium (Fig. 10 (1000)) comprising a plurality of executable instructions (Fig. 10 (1002)) which, when executed, implement an application program interface (API) (Fig. 10 (1004)) to dynamically generate one or more filter control command data structures in response to a command received from a decoder application (Fig. 2 (204); Page 52, line 19 through page 53, line 2; Fig. 3 (300)), wherein the one or more filter control command data structure(s) include one or more parameters which, when received by a communicatively coupled accelerator, effect one or more filter settings on the

accelerator in accordance with the received command (Page 23, lines 9-25; Fig. 2 (204); Page 62, lines 1-16; Fig. 10 (1000-1004)), wherein the API is not specific to any particular multimedia application and/or multimedia accelerator (Page 62, lines 11-12; page 15, lines 17-18).

With regard to claim 18, a computing system comprising: a decoder application (Fig. 2 (160 A-N)) to process received media content; and an operating system (Fig. 1 (158)) including an application program interface (API) (Fig. 1 (104); Fig 2 (104)), support the media processing, wherein the API generates one or more filter control commands including one or more parameters (Fig. 2 (204); Page 52, line 19 through page 53, line 2; Fig. 3 (300)) which, when received by a communicatively coupled media processing accelerator, effect one or more filter settings of the accelerator in accordance with a command received from the decoder (Page 23, lines 9-25; Fig. 2 (204)), wherein the decoder application is configured to iteratively issue configuration commands (Page 55, lines 1-4; Fig. 5 (502)) reflecting various alternative degrees and methods of decoding acceleration capability until choosing one that is acceptable to both the decoder application and the accelerator (Page 55, lines 1-10; Fig. 5 (502-508)).

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1-25 stand rejected under 35 U.S.C. §103(a) as being obvious over U.S. Patent No. 6,744,472 to MacInnis in view of U.S. Patent No. 6,725,279 to Richter, and further in view of U.S. Patent No. 6,725,279 to Sriram.

(7) Argument

A. The rejections under 35 U.S.C. §103(a) over MacInnis, Richter and Sriram fail because the Office has failed to establish a *prima facie* case of obviousness.

Applicant respectfully submits that the Office has not established a *prima facie* case of obviousness. The discussion below proceeds as follows. First, a section entitled “The § 103 Standard” is provided which describes the criteria that must be met in order to establish a *prima facie* case of obviousness. Second, a section entitled “The Claims” is provided which presents Applicant’s reasoning as to why the Office has not met these criteria.

The §103 Standard

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, ***there must be some suggestion or motivation***, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, ***to modify the reference or to combine reference teachings***. Second, there must be a reasonable expectation of success. Finally, ***the prior art reference (or references when combined) must teach or suggest all the claim limitations***. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on Applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The Claims

Claim 1 recites a method comprising:

- receiving a command from a decoder application at an application program interface (API), wherein the API is configured to facilitate the use of a plurality of different multimedia accelerators with the decoder application; and
- generating one or more filter control command data structures, recognizable by a communicatively coupled accelerator including one or more parameters which, when received by the accelerator, affects one or more filter settings of the accelerator based, at least in part, on the content of the received command.

In making out the rejection of this claim, the Office argues that its subject matter is rendered obvious in view of MacInnis, Richter, and Sriram. Specifically, the Office argues that MacInnis discloses all of the features of the claim except for an application interface (API). The Office then relies on Richter as disclosing an API and Sriram as further disclosing an API that is “configured to facilitate the use of a plurality of different multimedia accelerators with the decoder application”, as claimed. The Office argues that one would have been motivated to combine the teachings of these references “in order to obtain an apparatus that is more versatile by being able to perform complex operations.”

Applicant respectfully traverses this rejection and submits that the Office has not established a *prima facie* case of obviousness.

First, Applicant submits that the references do not collectively disclose all of the subject matter of this claim. For example, the *integrated circuit* system of MacInnis simply does not inherently disclose an application interface that would be necessary for it to correctly operate, as the Office contends. Also, Column 57,

lines 21-37, of MacInnis discuss the capabilities of the graphics accelerator, but not “generating one or more filter control command data structures”, as claimed. Furthermore, Sriram does not disclose or suggest an *API* that is configured to “facilitate”, as that term is used and understood in the context of the subject application (see e.g. Applicant’s disclosure, Page 15, line 2 through Page 16, line 9). In fact, the monitor processor in Sriram is part of the decoder itself (see e.g. Sriram, Fig. 1) and is simply a processor that splits picture decoding into multiple sub-processes based upon one of two schemes to maximize scalability and efficiency (see e.g. Sriram, Column 7, lines 49-50 and Column 8, lines 19-21).

Second, Applicant respectfully submits that the Office’s stated motivation “to obtain an apparatus that is more versatile by being able to perform complex operations” is too general and could serve as the basis for making *any* modification to MacInnis. Furthermore, this stated motivation is simply inapplicable to MacInnis and Sriram. Specifically, MacInnis teaches a graphics integrated circuit chip for controlling a television display. Furthermore, Sriram teaches an apparatus for decoding a MC-DCT video stream. Hence, the Office’s stated motivation simply makes no sense with respect to these references – each of which is concerned with specific operations that they can already perform. For instance, MacInnis teaches an integrated circuit for controlling a television display by processing “analog video input, digital video input, graphics input and audio input simultaneously”. Indeed, this is accomplished in a highly defined and ordered way. Accordingly, “to obtain an apparatus that is more versatile” by performing complex operations is simply not germane.

Finally, Applicant submits that modifying the integrated chip structure of MacInnis with the multiple bus managing application interface of Richter and the hierarchically regimented decoding system of Sriram would impermissibly render MacInnis unsatisfactory for its intended purpose and impermissibly change its principle of operation. (see MPEP 2143.01). Specifically, the application interface of Richter is designed to create a subset of processing blocks for a task in an environment comprising different buses and different communication protocols. (see Richter, Column 1). To do this, the interface selects blocks, examines each processing block's input and output interfaces to see if data exchange is possible, and modifies the selection of blocks if it is not. In stark contrast, MacInnis teaches a graphics display system contained in an *integrated circuit*. (see MacInnis, Fig. 1). This system includes a graphics display pipeline and video display, each containing defined functional blocks which process data in a specific order. (see e.g. MacInnis, Fig. 4). Accordingly, Richter's interface would provide no advantage to MacInnis, and would actually interfere with its well defined and orderly display pipelines. In fact, in so far as MacInnis teaches designated functional blocks which process data in a particular order, it teaches directly away from Richter's block-modifying interface.

Additionally, Sriram teaches a video decoder which includes multiple sub-processors and a monitor processor to control the sub-processors. (see e.g. Sriram, Fig. 1). In contrast, the video decoder in MacInnis does not itself comprise the accelerator or the memory controller. Indeed, the architectures of Sriram and MacInnis are so different, that even a cursory inspection shows that modifying McInnis would radically change its principle of operation and render it

unsatisfactory as an integrated chip for controlling a television display. Simply put, the teachings of MacInnis, Richter and Sriram are too technologically inconsistent for one to have been motivated to combine them.

In view of the above discussion, the Office has not established a *prima facie* case of obviousness. Accordingly, for at least this reason, Applicant traverses this rejection and submits that this claim is allowable.

Claims 2-11 depend from claim 1 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 1, are neither disclosed nor suggested in the references of record, either singly or in combination with one another.

In addition, regarding claim 3, the Office argues that Fig. 28 of MacInnis discloses “wherein the filter is a post-processing filter.” Applicant, after reviewing Fig. 28, fails to see how this subject matter is disclosed or suggested. Specifically, Fig. 28 deals with blending by the video compositor, not the accelerator. (see e.g. MacInnis, Column 44, lines 23-38 and Fig. 2). Accordingly, Applicant submits that the Office’s reliance on this figure is misplaced.

In addition, regarding claim 4, the Office argues that the subject matter of this claim is disclosed on Column 4 (lines 29-31) of Richter. Furthermore, the Office states that “filters and prediction references are well known within the environment of encoders and decoders”. Applicant respectfully disagrees. First, this excerpt simply does not disclose: “wherein output data subsequent to the application of a post-processing filter are used as prediction references”, as claimed. Second, the Office has not substantiated its claim that filters and

prediction references were well known. This excerpt from Column 4 is reproduced below for the Office's convenience:

This architecture is particularly used to implement very complex multimedia processing configurations using, for example, echo suppressors or encoding or decoding blocks with a very simple application interface.

In addition, regarding claims 6-7, the Office cites Column 4 (lines 40-51) of MacInnis and argues: "the strength parameter is the scaling". Applicant respectfully disagrees and submits that the Office has mischaracterized the MacInnis reference. Specifically, this cited excerpt merely discusses a video scaler that is responsible for scaling video stream input. This simply has nothing to do, whatsoever, with "a strength parameter to control an amount of filter applied by a receiving filter" or any other parameter(s) "which, when received by the accelerator, affects one or more filter settings of the accelerator", as claimed.

Furthermore, even if "the strength parameter was the scaling", which it is not, the Office appears to have forgotten that it relies (albeit inappropriately) on the "blending, scaling, blitting, and filling" of MacInnis's *graphics accelerator, not its video scaler*, for disclosing "the filter parameters". As such, the scaling performed by the video scaler in MacInnis could not possibly disclose the strength parameter recited in this claim. In addition, even a cursory glance at Fig. 2 of MacInnis illustrates that the video scaler 52 is completely separate from the accelerator 64 and performs a completely different function(s).

Claim 12 recites a storage medium comprising a plurality of executable instructions which, when executed, implement an application program interface

(API) to dynamically generate one or more filter control command data structures in response to a command received from a decoder application, wherein the one or more filter control command data structure(s) include one or more parameters, which, when received by a communicatively coupled accelerator, effect one or more filter settings on the accelerator in accordance with the received command, wherein the API is not specific to any particular multimedia application and/or multimedia accelerator.

In making out the rejection of this claim, the Office argues that its subject matter is rendered obvious in view of MacInnis, Richter, and Sriram. Specifically, the Office argues that MacInnis discloses all of the features of the claim except for an application interface (API). The Office then relies on Richter as disclosing an API and Sriram as further disclosing an API that is “configured to facilitate the use of a plurality of different multimedia accelerators with the decoder application”, as claimed. The Office argues that one would have been motivated to combine the teachings of these references “in order to obtain an apparatus that is more versatile by being able to perform complex operations.”

Applicant respectfully traverses this rejection and submits that the Office has not established a *prima facie* case of obviousness. First, as discussed above, Applicant submits that the references do not collectively disclose all of the subject matter of this claim. Second, also as discussed above, Applicant submits that the Office’s stated motivation is too general and could serve as the basis for making *any* modification to MacInnis. In addition, this stated motivation is simply inapplicable to MacInnis because MacInnis and Sriram are concerned with specific operations that they can already perform.

Furthermore, Applicant submits that modifying the integrated graphics chip structure of MacInnis with the hierarchically regimented decoding system of Sriram and the multiple bus architecture of Richter would impermissibly render MacInnis unsatisfactory for its intended purpose and impermissibly change its principle of operation. Specifically, as discussed in detail above, the application interface of Richter is designed to create a subset of processing blocks for a task in an environment comprising different buses and different communication protocols. (see Richter, Column 1). To do this, the interface selects blocks, examines each processing block's input and output interfaces to see if data exchange is possible, and modifies the selection of blocks if it is not. This is in stark contrast to the graphics display system of MacInnis. Furthermore, Sriram teaches a video decoder which includes various sub-processors and a monitor processor to control the sub-processors. (see e.g. Sriram, Fig. 1). In contrast, the video decoder in MacInnis does not itself comprise the accelerator or the memory controller. Simply put, the teachings of MacInnis, Richter and Sriram are too technologically inconsistent for one to have been motivated to combine them.

Finally, Applicant notes that the Office has not addressed "wherein the API is not specific to any particular multimedia application and/or multimedia accelerator." This is not surprising since the monitor processor (considered an API by the Office) in Sriram is actually part of the video decoder itself, which also includes all of the associated sub-processors (which are considered accelerators by the Office).

In view of the above discussion, the Office has not established a *prima facie* case of obviousness. Accordingly, for at least this reason, Applicant traverses this rejection and submits that this claim is allowable.

Claims 13-17 depend from claim 12 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 12, are neither disclosed nor suggested in the references of record, either singly or in combination with one another.

In addition, regarding claim 17, the Office cites Column 4 (lines 40-51) of MacInnis and argues: “the strength parameter is the scaling”. Applicant respectfully disagrees and submits that the Office has mischaracterized the MacInnis reference. Specifically, this excerpt merely discusses a video scaler that is responsible for scaling video stream input. This simply has nothing to do, whatsoever, with “a strength parameter to control an amount of filter applied by a receiving filter” or any other parameter(s) “which, when received by the accelerator, affects one or more filter settings of the accelerator”, as claimed.

Furthermore, even if “the strength parameter was the scaling”, which it is not, the Office appears to have forgotten that it relies (albeit inappropriately) on the “blending, scaling, blitting, and filling” of MacInnis’s *graphics accelerator, not its video scaler*, for disclosing “the filter parameters”. As such, the scaling performed by the video scaler in MacInnis could not possibly disclose the strength parameter recited in this claim. In addition, even a cursory glance at Fig. 2 of MacInnis illustrates that the video scaler 52 is completely separate from the accelerator 64 and performs a completely different function(s).

Claim 18 recites a computing system comprising:

- a decoder application to process received media content; and
- an operating system including an application program interface (API), support the media processing, wherein the API generates one or more filter control commands including one or more parameters which, when received by a communicatively coupled media processing accelerator, effect one or more filter settings of the accelerator in accordance with a command received from the decoder, wherein the decoder application is configured to iteratively issue configuration commands reflecting various alternative degrees and methods of decoding acceleration capability until choosing one that is acceptable to both the decoder application and the accelerator.

In making out the rejection of this claim, the Office argues that its subject matter is rendered obvious in view of MacInnis, Richter, and Sriram. Specifically, the Office argues that MacInnis discloses all of the features of the claim except for an application interface (API). The Office then relies on Richter as disclosing an API and Sriram as further disclosing an API that is “configured to facilitate the use of a plurality of different multimedia accelerators with the decoder application”, as claimed. The Office also relies on Columns 5 and 6 of Sriram as disclosing “wherein the decoder application is configured to iteratively issue configuration commands reflecting various decoder acceleration capabilities until choosing one that is acceptable to both the decoder application and the accelerator”. The Office then argues that one would have been motivated to combine the teachings of these references “in order to obtain an apparatus that is more versatile by being able to perform complex operations.”

Applicant respectfully traverses this rejection and submits that the Office has not established a *prima facie* case of obviousness.

First, as discussed above, Applicant submits that the references do not collectively disclose all of the subject matter of this claim. In addition, with respect to Columns 5 and 12 of Sriram, Applicant submits that the Office has mischaracterized these excerpts. Specifically, these portions simply have nothing to do with a decoder application that is “configured to *iteratively* issue configuration commands reflecting various *alternative degrees and methods of decoding acceleration capability until choosing one that is acceptable to both the decoder application and the accelerator*”. (emphasis added). These excerpts are reproduced below for the Office’s convenience:

Data structures are one component of the invention. Data structures for different block communication and parameter passing have been chosen according to the bit stream hierarchy. Several factors were considered in determining the organization of these parameters. Some of the factors are: (1) implementing video decoding using multiple processes efficiently; (2) efficient argument passing between different compute blocks; (3) computational efficiency; (4) efficient data flow (minimal data replication); and (5) good data cache effects.

Any given processing unit (for example, Motion Compensation) needs only a subset of these parameters. A Macroblock structure is defined in such a way that all the parameters needed for Macroblock processing can be found in the MB data structure.

The Office’s only argument with respect to these excerpts is to state “wherein the configuration commands is the parameter passing”. Applicant respectfully submits that this explanation is insufficient because it fails to account for all of the language of this element.

Second, also as discussed above, Applicant submits that the Office's stated motivation is too general and could serve as the basis for making *any* modification to MacInnis. In addition, this stated motivation is simply inapplicable to MacInnis because MacInnis and Sriram are concerned with specific operations that they can already perform.

Finally, Applicant submits that modifying the integrated graphics chip structure of MacInnis with the hierarchically regimented decoding system of Sriram and the multiple bus architecture of Richter would impermissibly render MacInnis unsatisfactory for its intended purpose and impermissibly change its principle of operation. Specifically, as discussed in detail above, the application interface of Richter is designed to create a subset of processing blocks for a task in an environment comprising different buses and different communication protocols. (see Richter, Column 1). To do this, the interface selects blocks, examines each processing block's input and output interfaces to see if data exchange is possible, and modifies the selection of blocks if it is not. This is in stark contrast to the graphics display system of MacInnis. Furthermore, Sriram teaches a video decoder which includes various sub-processors and a monitor processor to control the sub-processors. (see e.g. Sriram, Fig. 1). In contrast, the video decoder in MacInnis does not itself comprise the accelerator or the memory controller. Simply put, the teachings of MacInnis, Richter and Sriram are too technologically inconsistent for one to have been motivated to combine them.

In view of the above discussion, the Office has not established a *prima facie* case of obviousness. Accordingly, for at least this reason, Applicant traverses this rejection and submits that this claim is allowable.

Claims 19-25 depend from claim 18 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 18, are neither disclosed nor suggested in the references of record, either singly or in combination with one another.

In addition, regarding claim 20, the Office argues that Fig. 28 of MacInnis discloses “wherein the filter is a post-processing filter.” Applicant, after reviewing Fig. 28, fails to see how this subject matter is disclosed or suggested. Specifically, Fig. 28 deals with blending by the video compositor, not the accelerator. (see e.g. MacInnis, Column 44, lines 23-38 and Fig. 2). Accordingly, Applicant submits that the Office’s reliance on this figure is misplaced.

In addition, regarding claim 23, the Office cites Column 4 (lines 40-51) of MacInnis and argues “the strength parameter is the scaling”. Applicant respectfully disagrees and submits that the Office has mischaracterized the MacInnis reference. Specifically, this excerpt merely discusses a video scaler that is responsible for scaling video stream input. This simply has nothing to do, whatsoever, with “a strength parameter to control an amount of filter applied by a receiving filter” or any other parameter(s) “which, when received by the accelerator, affects one or more filter settings of the accelerator”, as claimed.

Furthermore, even if “the strength parameter was the scaling”, which it is not, the office appears to have forgotten that it relies (albeit inappropriately) on the “blending, scaling, blitting, and filling” of MacInnis’s *graphics accelerator, not its video scaler*, for disclosing “the filter parameters”. As such, the scaling performed by the video scaler in MacInnis could not possibly disclose the strength

parameter recited in this claim. In addition, even a cursory glance at Fig. 2 of MacInnis illustrates that the video scaler 52 is completely separate from the accelerator 64 and performs a completely different function(s).

Conclusion

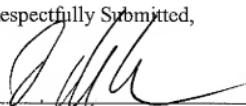
The Office has failed to establish a *prima facie* case of obviousness. Accordingly, Applicant respectfully requests that the rejections be overturned and that the pending claims be allowed to issue.

Dated:

6/14/2006

Respectfully Submitted,

By:


Rich Bucher
Lee & Hayes, PLLC
Reg. No. 57,971
(509) 324-9256 ext. 216

(8) Appendix of Appealed Claims

1. (Previously Presented) A method comprising:
receiving a command from a decoder application at an application program interface (API), wherein the API is configured to facilitate the use of a plurality of different multimedia accelerators with the decoder application; and
generating one or more filter control command data structures, recognizable by a communicatively coupled accelerator including one or more parameters which, when received by the accelerator, affects one or more filter settings of the accelerator based, at least in part, on the content of the received command.
2. (Original) A method according to claim 1, further comprising:
passing the generated filter control command data structures to the accelerator, wherein the accelerator modifies one or more filter settings in accordance with the parameters embedded within the data structure.
3. (Original) A method according to claim 1, wherein the filter is a post-processing filter.
4. (Original) A method according to claim 3, wherein output data subsequent to the application of a post-processing filter are used as prediction references for decoding subsequent data.

5. (Original) A method according to claim 3, wherein the post-processing filter is one or more of a deblocking filter, a de-ringing filter, and the like.

6. (Original) A method according to claim 1, wherein the parameters include a strength parameter.

7. (Original) A method according to claim 6, wherein the generated data structure includes a strength parameter for each of one or more block boundaries of a frame.

8. (Original) A method according to claim 1, wherein the API issues filter control commands for each of a number of edges of luminance and chrominance blocks of received media content.

9. (Original) A method according to claim 1, wherein the API issues macroblock filter control command data structures for each macroblock of video picture content, each macroblock filter control command comprising four (4) or sixteen (16) luminance block filter control command data structures for controlling the filtering of the luminance blocks of the macroblock, and/or two (2), four (4), eight (8), sixteen (16), or thirty-two (32) chrominance block filter control command data structures for controlling the filtering of the chrominance blocks of the macroblock.

10. (Original) A storage medium comprising a plurality of executable instructions which, when executed, implement a method according to claim 1.

11. (Original) A computing system comprising:
a storage medium having stored therein a plurality of executable instructions; and
an execution unit, coupled to the storage medium, to execute at least a subset of the plurality of executable instructions to implement a method according to claim 1.

12. (Previously Presented) A storage medium comprising a plurality of executable instructions which, when executed, implement an application program interface (API) to dynamically generate one or more filter control command data structures in response to a command received from a decoder application, wherein the one or more filter control command data structure(s) include one or more parameters which, when received by a communicatively coupled accelerator, effect one or more filter settings on the accelerator in accordance with the received command, wherein the API is not specific to any particular multimedia application and/or multimedia accelerator.

13. (Original) A storage medium according to claim 12, wherein the filter control command data structure(s) effect one or more post processing filter(s) of the accelerator.

14. (Original) A storage medium according to claim 12, wherein the filter control command data structure(s) effect one or more of a deblocking filter(s), dc-ringing filter(s), and/or another post processing filter of the accelerator.
15. (Original) A storage medium according to claim 12, wherein the API issues a filter control command data structure for each of a number of edges of luminance and chrominance blocks of received media content.
16. (Original) A storage medium according to claim 15, wherein the API issues four (4) filter control command data structures for each luminance block, and/or two (2) filter control command data structure(s) for each chrominance block.
17. (Original) A storage medium according to claim 12, wherein the parameter(s) include a filter strength parameter.
18. (Previously Presented) A computing system comprising:
a decoder application to process received media content; and
an operating system including an application program interface (API), support the media processing, wherein the API generates one or more filter control commands including one or more parameters which, when received by a communicatively coupled media processing accelerator, effect one or more filter settings of the accelerator in accordance with a command received from the

decoder, wherein the decoder application is configured to iteratively issue configuration commands reflecting various alternative degrees and methods of decoding acceleration capability until choosing one that is acceptable to both the decoder application and the accelerator.

19. (Original) A computing system according to claim 18, further comprising:

one or more media processing accelerator(s), communicatively coupled to the decoder application via the API, including one or more filter(s) responsive to the filter control command data structures reflecting information received in the command from the decoder.

20. (Original) A computing system according to claim 19, wherein the filter(s) are post processing filters.

21. (Original) A computing system according to claim 19, wherein the filter(s) include one or more of a deblocking filter, de-ringing filter, and the like.

22. (Original) A computing system according to claim 18, wherein the API issues macroblock filter control command data structures for each macroblock of video picture content, each macroblock filter control command comprising four (4) or sixteen (16) luminance block filter control command data structures for controlling the filtering of the luminance blocks of the macroblock, and/or two (2), four (4), eight (8), sixteen (16) or thirty-two (32) chrominance block filter control command data structures for controlling the filtering of the chrominance blocks of the macroblock.

23. (Original) A computing system according to claim 18, wherein the filter control command data structures include a strength parameter to control an amount of filter applied by a receiving filter.

24. (Original) A computing system according to claim 18, further comprising:
a storage medium having stored therein a plurality of executable instructions; and
an execution unit, coupled to the storage medium, to execute at least a subset of the plurality of executable instructions to implement the operating system and associated API.

25. (Original) A computing system according to claim 24, wherein the execution unit executes at least a subset of the plurality of executable instructions to implement the decoder application.

(9) Evidence appendix. None

(10) Related proceedings appendix. None